

The RTEMS RTOS for EPICS Controls at SPEAR

Stephanie Allison
Clemens Wermelskirchen
Harvey Rarback
Till Straumann
SSRL/SLAC

Overview

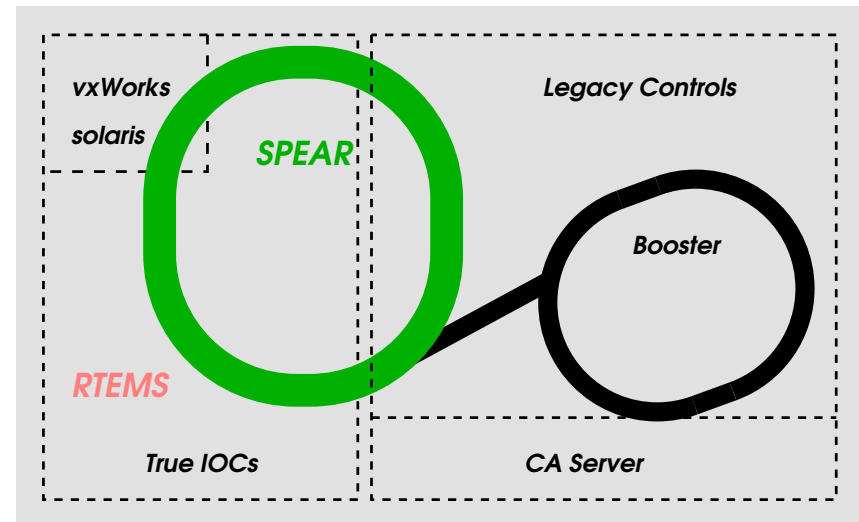
- Spear Control System Overview
- Why Do We Need a Real-Time OS
- What Is RTEMS?
- How SPEAR Changed Horses
- Production Software Environment
- Demo
- Questions

The Spear Control System

3rd Generation light source
SW development 2000-2004
In situ replacement, summer 2003

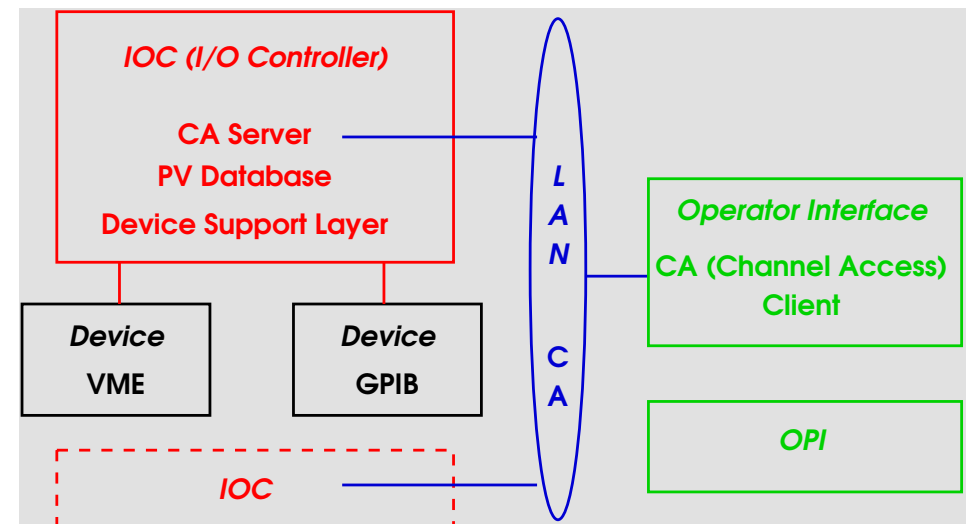
Controls:

- Legacy control system (Camac, VMS) with CA server for booster and many Spear slow controls
- 25 EPICS IOCs running RTEMS
- 1 EPICS/vxWorks IOC controls RF station (PEP2 technology, aka "black box")
- 1 "Soft" IOC on Solaris



Experimental Physics and Industrial Control System (EPICS)

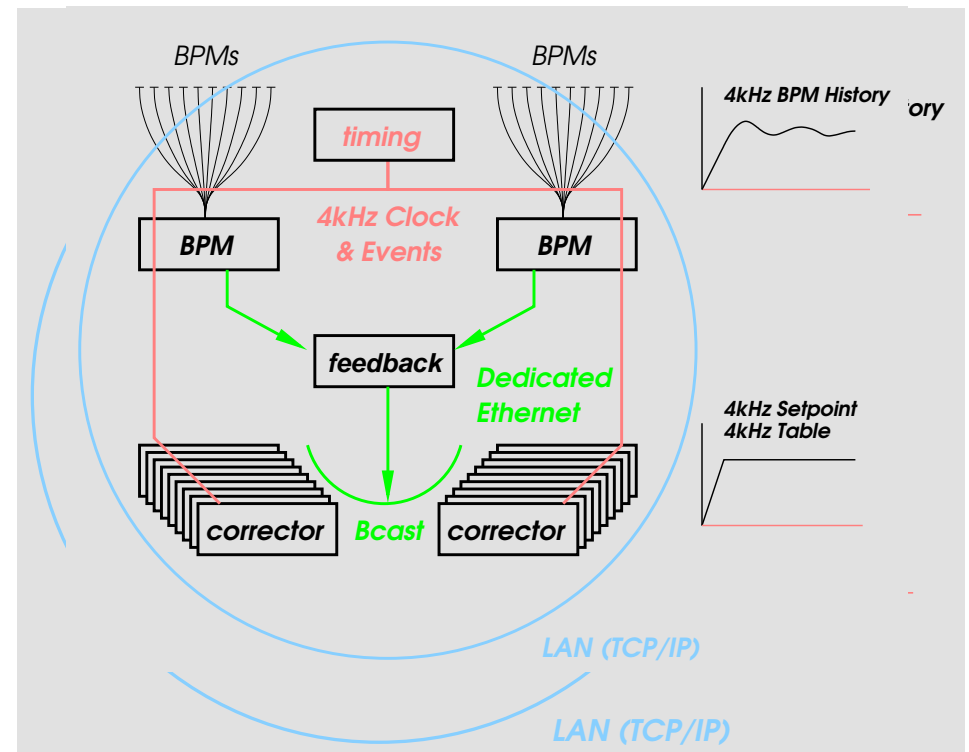
- Distributed Control System
- I/O Controller (IOC)
 - Drives devices
 - Process Variable (PV) abstraction; "database"
 - Db server; Channel Access Protocol (CA)
- CA Protocol
 - Name service
 - Data typing
 - Display information
 - Alarm / status
 - Timestamps
- Operator Interfaces
 - Channel Access Clients.
 - Lookup PVs by name
 - Read/write PVs
 - GUI, matlab, java, scripting languages



Spear IOC Applications

Key Applications are fast orbit controls (eventually feedback):

- System clock rate: 4Khz
- Distributed:
 - 96 Beam position monitors (currently 57) are processed in two remote locations (2 IOCs)
 - 108 Fast corrector magnets are controlled by 18 IOCs
 - 4kHz clock & timestamps distribution
 - Synchronous capture of diagnostic waveforms; available over CA
 - 4kHz streamed data communication over dedicated ethernet
 - Central Orbit Feedback Processor (2004)



Real-Time Software Architecture

- Basic idea: run EPICS on RTOS. High priority task does hard RT work and uses EPICS for I/O.
- Timestamp/Event receiver synchronizes high priority real-time task.
- RT task decimates data, fills fifos, drives setpoint tables etc.
- I/O: EPICS "Device support" connects RT layer with EPICS database. RT-features integrate nicely with EPICS.
- The Channel Access protocol delivers timestamped slow PVs, history buffers and loads setpoint tables etc.
- Performance determined by underlying RTOS. EPICS provides easy access and short development.

Selecting a real-time OS for SPEAR

- Design started in 2000 when an OSI version of EPICS was released
- Evaluation of different "Open Source" RTOSes; RTEMS / RTLinux
- Open Source:
 - Long lifecycle needed; access to source (OS + tools) is a warranty for longevity (even if "vendor" drops support!!)
 - Ultimate documentation (writing drivers; understanding quirks)
 - Self support; fixing bugs often faster than reporting them
 - No license hassles
 - Bigger variety of host platforms
- Choice of RTEMS (clean design, better performance; see ICALEPCS 2001)

Strategy Issues for SPEAR

□ Pioneering Risks

- RTEMS/EPICS had only been used at CLS; no VME, no production release, probably no RT app.
- Limited resources of SPEAR3 (but more flexible), few people

□ Strategy

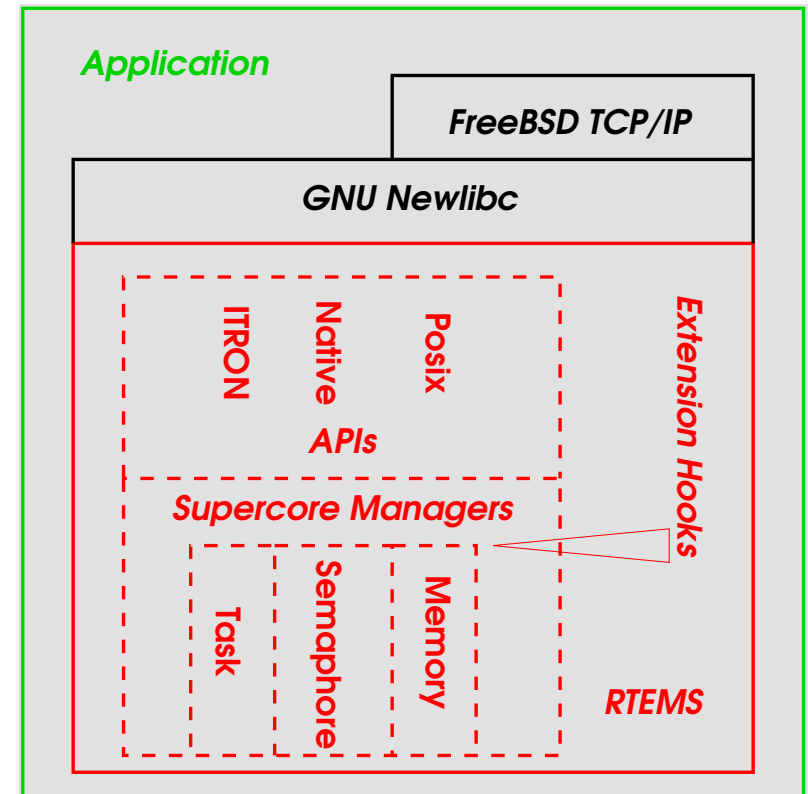
- Evaluation
- Developing / porting to EPICS' OSI layer rather than RTEMS native API
- Use few, standard features not exotic ones
- Leave escape route
- Design Reviews

What is RTEMS?

- ❑ Hard real-time "kernel"; developed by OAR Corp. in the late 80s on behalf of the US Army Missile Research and Development Center
- ❑ Real Time Executive for ~~Missile Military~~ Multiprocessor Systems
- ❑ Open source
- ❑ Commercial support available (E.g., APS has OAR develop a MVME2100 BSP)
- ❑ Active mailing list, good on-line documentation

RTEMS Architecture

- Modular, sleek and cleanly written design (core)
- Multiple APIs, Posix (pthreads), native, ITRON
- Extension Hooks (e.g., callback at task switch)
- Library, linked to application
- Configurable (e.g., # of semaphore, managers to use,...)
- Many CPU Architectures (ix86, powerpc, 68k, ...)
- GNU Newlibc (reentrant) C-library
- C++ support
- Filesystems: IMFS (ramdisk), others (DOSfs - not used by SSRL)

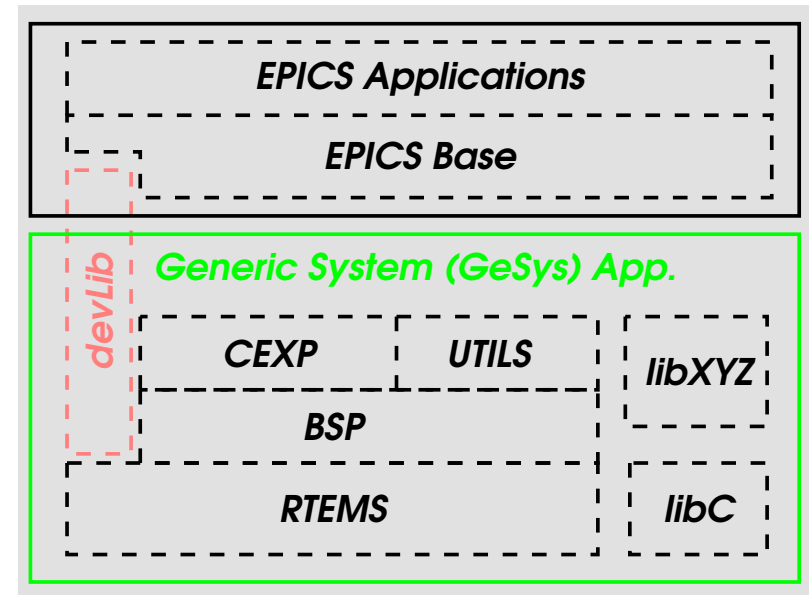


RTEMS Networking

- FreeBSD TCP/IP stack
- Sun RPC/XDR library
- Simple implementations for
 - NFS client
 - NTP client
 - TELNET server
- TFTP 'filesystem'
- Others (pppd, www, ftp - not used at SSRL)

SPEAR RTEMS Software Development Roadmap 2000-2004

- EPICS 3.14 beta / OSI; RTEMS port (available 2000)
- Performance Evaluation (ICALEPCS 2001)
- BSP (Synergy Microsystems VGM series SBC)
- Design Review (2002)
- Core software package (coll. meetings 2002)
 - Cexp shell (C expressions) and run-time loader/linker;
 - automatic symbol table access
 - trivial (nested) scripting
 - NFS client
 - Misc: (NTP client, telnet server, coredump, monitor)
 - Generic system application provides run-time environment a la vxWorks target shell (LDEP config utility)
- Available for PPC, PCx86, m68k (MVME167)
- devLib implementation makes porting ("3.14ification") existing drivers easy (IPAC, GTR), provides abstraction from RTEMS.



RTEMS Development

- GNU toolchain (C, C++, Fortran, Assembler)
- EPICS build system
- Source / version control (CVS)
- AFS (host filesystem)
- Typical development cycle (non-EPICS)
 - compile
 - load/link into running target
 - test
 - unload/unlink
- EPICS core software cannot be stopped - reboot still needed.

Spear Production RTEMS Environment

Modular (eases maintenance, comfortable):

- Generic platform for all IOCs (GeSys)
- BOOTP/Netboot over TFTP (rsh, NFS) (minimize NVRAM settings)
- Generic add-on utilities (telnetd, ...)
- Multi-level scripts (generic system - IOC groups - IOC specific, ...) for loading EPICS applications & databases.
- Using C-expressions from scripts helps, e.g.:
 - `probeDriver("device") && dbLoadRecords("dbPiece.db","xxx")`

Main Applications

- RT for orbit control & feedback
- Slow bitbus power supply control
- IP modules (digital I/O, CANbus, Hytek ADC)
- PMC modules (64 channel 16bit parallel ADC, Ethernet, Race++)
- GPIB (various) over LAN
- VME boards (Echotek digital receiver, Joerger VTR, BitBus, Spear Timestamp)
- Proprietary (MCOR30) corrector power supply controller

Number of SPEAR Process Variables

- 25 RTEMS IOCs in production; 21 run RT applications at 4kHz
- PVs are mirrored in a RDB database and setpoints restored across reboots.
- 4600 PVs constantly monitored and logged (excluding legacy/VMS system)
- 32000 records alive on IOCs
- Longest task dispatching latency recorded ~50us (330MHz PPC)

Overall Experience

- In production on ~25 IOCs
- Very reliable
- Good RT performance (4kHz tasks on 20 IOCs never missed schedule)
- Transparent to avg. EPICS developer
- Found 1 bug -- fixed within 1-2 days
- No licensing hassle, no fees
- Total control over fixing problems
- Fun Factor

Demonstration

- Boot a generic RTEMS system on MVME2307
- Demonstrate the CEXP shell
- Write, load and run 'hello_world'
- Start EPICS example IOC

- Evaluation CD available at
 - <http://www.slac.stanford.edu/~strauman/rtems/cd/index.html>